

PD3

①日本国特許庁  
公開特許公報

①特許出願公開  
昭53—110338

⑤Int. Cl.<sup>2</sup>  
G 06 F 5/02

識別記号

⑥日本分類  
97(7) E 21

庁内整理番号  
7165—56

④公開 昭和53年(1978)9月27日

発明の数 1  
審査請求 有

(全 18 頁)

④符号化装置

①特 願 昭53—23698

②出 願 昭53(1978)3月3日

優先権主張 ③1977年3月4日③アメリカ国  
(US)④774365

⑦発 明 者 グレシ・ジョージ・ラングドン  
・ジュニア  
アメリカ合衆国カリフォルニア  
州サン・ホセ・ハンプトン・ド  
ライブ6672番地

⑦発 明 者 ジョーマ・ジョーハネン・リザ  
ネン  
アメリカ合衆国カリフォルニア  
州サン・ホセ・アドニス・ウエ  
イ2442番地

⑦出 願 人 インターナショナル・ビジネス  
・マシーンス・コーポレーショ  
ン  
アメリカ合衆国10504ニューヨ  
ーク州アーモンク(番地なし)

⑦復代理人 弁理士 頓宮孝一

明 細 書

1. 発明の名称 符号化装置

2. 特許請求の範囲

(1) 任意の順序で番号付けられたN個のシンボル  
 $a_1$  乃至  $a_N$  を含む情報源から送られてきたシン  
ボル列を符号化する際に、該シンボル列中の各シン  
ボル毎に順次符号化操作を行なつて、その度に  
新しい符号列を生成する符号化装置にして、

上記シンボル列中の各シンボル  $a_k$  に対応して、  
その生起確率  $p(a_k)$  によつて決まる長さ表示  
 $l(a_k)$  を引出すための第1メモリ手段と、

上記長さ表示  $l(a_k)$  から、新しい符号列の  
長さ表示  $L$  の整数部分  $y$  及び小数部分  $x$  を計算す  
るための演算手段と、

上記小数部分  $x$  と、上記情報源におけるシンボ  
ルの番号付けに従う累積生起確率  $P_{k-1}$  とによ  
つて決まる  $r$  デイジットの値  $A_k$  を発生するため  
の第2メモリ手段と、

先行符号列の上位  $r$  デイジットのうちの低位の

$y$  デイジットをそのまま出力するためのシフト手段と、

先行符号列の上位の  $r - y$  デイジットと上記値

$A_k$  とを加算して新しい符号列の上位  $r$  デイジッ  
トを生成するための加算手段とを有する符号化装

置。  
(2) 上記長さ表示  $l(a_k)$  の値は、 $\sum_{k=1}^N 2^{-l(a_k)} \leq 1$   
によつて制限される特許請求の範囲第(1)項記載の  
符号化装置。

(3) 上記演算手段は、各符号化操作において先行  
符号列の長さ表示  $L'$  と上記長さ表示  $l(a_k)$   
との和から上記整数部分  $y$  及び上記小数部分  $x$  を  
得る特許請求の範囲第(2)項記載の符号化装置。

(4) 上記演算手段は、各符号化操作において、先  
行符号列の長さ表示  $L'$  の小数部分  $x'$  と上記長  
さ表示  $l(a_k)$  との和から上記整数部分  $y$  及び  
上記小数部分  $x$  を得る特許請求の範囲第(2)項記載  
の符号化装置。

(5) 上記第2メモリ手段によつて発生される値  $A_k$   
は、 $P_{k-1} \cdot 2^x$  を  $r$  デイジットで近似したも  
のである特許請求の範囲第(1)項、第(2)項、第(3)項

又は第(4)項記載の符号化装置。但し、 $P_{k-1} = \sum_{i=1}^{k-1} p(a_i)$ 、 $P_0 = 0$ 、 $P_N = 1$ である。

(6) 上記シフト手段は、先行符号列の上位  $r$  デジタルを  $y$  デジタル分だけ右シフトさせる右シフト装置から成っている特許請求の範囲第(5)項記載の符号化装置。

(7) 上記第1メモリ手段及び上記第2メモリ手段は、上記情報源に含まれるすべてのシンボルについて予め計算された  $L(a_k)$  及び  $A_k$  ( $k=1, 2, \dots, N$ ) を記憶している特許請求の範囲第(1)項乃至第(6)項の1つに記載の符号化装置。

### 3. 発明の詳細な説明

本発明は、有限の情報源アルファベットからのシンボル列の符号化（短縮 (compaction) を行なうため、シンボルの相対生起頻度を利用するもの）及び復号に関するものであり、特に可算 (enumerative) 符号化法及びハフマン短縮法の改良に関するものである。

ハフマン短縮法（符号化法）は情報源アルファベットの各文字に符号ワードを割当てるときに、こ

ら、これは、符号ワードが先験シンボルの固定シーケンスから構成され、すべてのワードが区別でき（非特異性）、そしてどの符号ワードも別の符号ワードのプレフィクスではない（一意復号可能）という古典符号の域を出ない。

可算符号化法は、例えば  $n$  個のシンボルから成る2進列の各置換を符号化するもので、まず  $2^n$  個の項を順番に並べて、各項を序数で表示することにより行なわれる。可算符号化法の詳細については、1973年1月に発行された IEEE Transactions on Information Theory の73乃至77頁に掲載されているコーバー (T. M. Cover) の論文“可算情報源符号化法 (Enumerative Source Encoding)”を参照されたい。簡単に説明すると、例えば  $n$  個の2進デジタルのうち1が  $m$  個ある2進列の集合は、 $T(n, m)$  で表わされる。項は、一貫した方法で、即ち予め決めた優先順位（左から右）に基づいて大きさによつて（辞書式順序）順番に並べる必要がある。例えば、 $n=3$  及び  $m=2$  と

の符号ワードの長さを文字の相対生起頻度に反比例させるものである。生起統計は、静的又は準静的であつた。即ち、この方法は、一定の適応性ある符号ワードの再配列及び再割当てを考慮していなかつた。

1952年9月に発行された Proceedings of the IRE の1098乃至1101頁に記載されているハフマンの最初の論文“最小冗長符号の構成方法 (A Method for the Construction of Minimum Redundancy Codes)”以来、例えば米国特許第3675211号明細書、同第3675212号明細書、同第3701111号明細書、同第3694813号明細書及び同第3717851号明細書に示されているように、幾多の修正がなされてきた。このような符号化法のバックグラウンドについては、1963年にマグロー・ヒル社から発行されたノーマン・アブラムソン著の“情報論理及び符号化 (Information Theory and Coding)”の第3章及び第4章を参照されたい。しかしなが

とすると、 $011 < 101 < 110$  のように順序付けてもよい。順序付けにおいて各項に位置番号を割当てると、最終項は2項係数  $\binom{n}{m} = n! / (n-m)! m!$  になる。他のすべての位置番号（符号ワード）はこれよりも小さい。

ハフマン符号化法及び可算符号化法においては、シンボル当りの達成可能な平均符号長は、情報源シンボルの頻度についてのエントロピー関数によつて決まる下限を有している。

情報源シンボル列  $s$  を  $s = a_1 a_2 \dots$ （右側に延びる）で表わし、これの終端（右端）に新しいシンボル  $a_k$  ( $k=1, 2, \dots, N$ ) を付加することによつて得られる新しいシンボル列を  $s a_k$  で表わすものとする。ハフマン符号化法においては  $s a_k$  の符号化表示は、 $s$  の符号の右端にシンボル  $a_k$  の符号ワード  $A_k$  を付加即ち連結することによつて得られる。これは、符号ワード長がシンボルの生起頻度とは無関係に付加シンボルのビット数だけ長くなることを意味する。言い換えれば、ハフマンの符号化は、2進情報源の列に

対しては短縮を行ない得ず、また少数のアルファベットに対しては不十分な短縮しか行なわない。

ハフマンの短縮法を改良するため、少数の情報源アルファベットに対して、ブロック化として知られている技法が提案された。ブロック化においては、新しい情報源アルファベットの新しい情報源シンボルは、元の2以上の情報源シンボルの並置(juxtaposition)である。一度にk個のシンボルをまとめてとりあげると、情報源アルファベットの大きさは $N^k$ まで増大する。

可算符号化は、このようなブロック化を必要としないが、新しい各符号ワード $C(s a_k)$ の計算には、元の符号ワード即ち符号列 $C(s)$ の全ビットについての情報が必要となる。従つて、限りなく増大する記憶容量を有するメモリが必要である。

本発明の目的は、ハフマン法におけるブロック化及び単純可算符号化を支持するための無限大の(限りなく増大する)メモリを必要とすることなく、符号列等に2進ビット列の符号化及び復号を行なうための装置を提供するにある。

まず長さ表示が決定され、次いでその小数部分及び整数部分を一意的に利用することにより新しい符号列が生成される。符号化を行なう場合には、新しい符号列のための長さ表示 $L(s a_k)$ が決定された後、新しい符号列が計算される。これは、次の帰納的数式(recurbion)によつて簡単に表わされる。

$$L(s a_k) = L(s) + L(a_k) \quad (1)$$

$$C(s a_k) = C(s) + P_{k-1} \cdot 2^{L(s a_k)} \quad (2)$$

上式において、 $L(s)$ は先行符号列 $C(s)$ のための長さ表示であり、 $L(a_k)$ は $p(a_k)$ を事前生起確率としたときの $\log_2 \{1/p(a_k)\}$ を有理数で近似したものである。シンボル $a_k$ に対応するこの長さ表示 $L(a_k)$ はクラフトの不等式:

$$\sum_{k=1}^N 2^{-L(a_k)} \leq 1 \quad (N \text{ はシンボル数})$$

を満足するものでなければならない。 $P_{k-1}$ は情報源アルファベットにおけるシンボルの任意の

本発明の他の目的は、シンボルの並べ方及び情報源アルファベットの大きさには実質的に無関係な算術列符号化法を用いる装置を提供するにある。これらの目的は、情報源シンボルの相対生起頻度が既知であるか又は推定可能であれば、達成することができる。

シンボルの列 $s = a_i a_j \dots$ を考えると、ハフマンの符号化によれば、有限の情報源アルファベットから取り出される一意的な各シンボルに対し、先験符号ワードが割当てられる。可算符号化においては、列 $s$ は符号化された列 $C(s)$ として表わされ、列 $s$ に次のシンボル $a_k$ が付加されると、新しい符号列 $C(s a_k)$ の生成には、先行符号列 $C(s)$ の全体を考慮する必要がある。

新しい符号列 $C(s a_k)$ は、情報源アルファベットのすべてのシンボルを任意に順序付けた場合の各シンボル $a_k$ の順序数即ち位置番号 $k$ と、長さ表示の小数部分及び整数部分と、先行符号列 $C(s)$ の最後の数ディジットとで表わされ得ることが判明した。この事実から、本発明においては、

順序付けに基づいた累積生起確率であつて、次式で表わされる。

$$P_{k-1} = \sum_{i=1}^{k-1} p(a_i), \quad P_0 = 0, \quad P_N = 1.$$

既に述べたように、本発明においては、長さ表示の小数部分及び整数部分が共に利用される。例えば、

$$\begin{aligned} L(s a_k) &= L(s) + L(a_k) \\ &= \text{整数 } y + \text{小数 } x \\ &= y(s a_k) + x(s a_k) \end{aligned}$$

において、これを(2)式に代入すると、次のようになる。

$$\begin{aligned} C(s a_k) &= C(s) + P_{k-1} \cdot 2^{L(s a_k)} \\ &= C(s) + P_{k-1} \{ 2^{y(s a_k)} \cdot 2^{x(s a_k)} \} \quad (3) \end{aligned}$$

$y$ は整数であるから、レジスタにおいて $2^{\pm y}$ を乗算することは、 $+y$ については左シフトを、 $-y$ については右シフトを各々行なうことになる。因数 $2^x$ は、 $q$ ビットで近似( $0(x) \approx 2^x$ )することができる。これはテーブル索引(table

look up)に役立つ。

これらの考察から、本発明の一実施例においては、新しい符号列は次のようにして生成される。

- (1)  $L(a_k)$  についてのテーブル索引
- (2)  $L(s a_k)$  を得るための加算
- (3)  $e(x) \approx 2^x$  についてのテーブル索引
- (4)  $P_{k-1}$  及び  $e(x)$  の乗算
- (5)  $y$  ビットの左シフト
- (6)  $C(a)$  及び  $P_{k-1} \cdot e(x) 2^y$  の加算

本発明の別の実施例では、ステップ(3)及び(4)が組合わされ、そしてステップ(5)では右シフトが行なわれる。これは、(3)式を次のように書き直したものに基いている。

$$2^{-y(s a_k)} C(s a_k) = 2^{-y(s a_k)} (x(a) + A_k)$$

但し

$$A_k = P_{k-1} 2^x(s a_k)$$

である。

- (2)  $L(s a_k) = y(s a_k) + x(s a_k)$  を得るために、 $x(a)$  と  $L(a_k)$  とを加算
- (3)  $A_k$  についてのテーブル索引
- (4)  $y(s a_k)$  ビットだけ  $C(a)$  を右シフト
- (5)  $A_k$  及び  $2^{-y(s a_k)} \cdot C(a)$  を加算

この手順によれば、新しい符号列の計算が簡単になり、シフトされた先行の符号列に  $A_k$  を加算するだけでよい。これは  $A_k$  が位置番号  $k$  及び小数  $x(s a_k)$  の関数である、という事実に基づいている。更にこの手順は、 $A_k$  及び  $C(a)$  が2進小数点を基準にして同じ相対位置を保持するということを利用している。この点について、 $P_{k-1}$  及び  $x(s a_k)$  が0と1の間にあることを想起されたい。

符号化においては、情報源シンボル列は符号化装置へ直列に入力され、符号化された2進列がそこから発生される。この符号化2進列は、その開始点及び終点を考慮することによつてのみ復号され得る。もし  $C(s a_k)$  が0と2の間の大きさ

特開昭53-110338(4)

上式によれば、先行符号列  $C(a)$  を  $y(s a_k)$  ビットだけ右シフトして  $A_k$  に加算すると、同じく  $y(s a_k)$  ビットだけ右シフトされた符号列  $C(s a_k)$  が得られる。ここで、 $y(s a_k)$  が新しい符号列  $C(s a_k)$  の長さ表示  $L(s a_k)$  の整数部分、即ち、(先行符号列  $C(a)$  の長さ表示  $L(a)$  + 付加シンボル  $a_k$  の長さ表示  $L(a_k)$ ) の整数部分であることを考えると、上式の右辺において  $C(a)$  を  $y(s a_k)$  ビットだけ右シフトさせることは、前回の計算で得られた  $2^{-y(a)} C(a)$  を、 $L(a)$  の小数成分  $x(a)$  と  $L(a_k)$  との和の整数成分の値だけ右シフトさせることと同等である。従つて、 $y(s a_k)$  として  $x(a) + L(a_k)$  の整数成分の値を用い、上式の左辺を新しく  $C(s a_k)$  とおくと、次式が得られる。

$$C(s a_k) = 2^{-y(s a_k)} C(a) + A_k \quad (4)$$

上式に従う符号化手順は次の通りである。

- (1)  $L(a_k)$  についてのテーブル索引

を有するものに制限されるならば、 $C(s a_k)$  の2進小数点は、最後の被加数  $A_k$  と特列され得る。従つて、 $C(s a_k)$  の左端(先頭)のビットが整数部分を表わし、その右側のビットが小数部分を表わす。

符号化装置の内部では、符号列は、 $r$  個のディジットから成っている。この " $r$  デイジット" は、 $A_k$  についての精度のインデックスでもある。 $C(a)$  の下位の  $y$  デイジットは、1サイクルの間に符号化装置から桁送りによつて出力され、上位デイジットの  $r - y$  デイジットが新しい計算において列に付加される。パラメータ " $r$ " は、生起頻度の最も低いシンボルの生起確率を  $p_m$  としたときに、 $\log(1/p_m)$  以上の値に設定されるのが好ましい。

復号においては、圧縮された(compressed)ビット列を復号可能な文字へ"解析(parse)"し、次いで符号化のときと同じような方法で文字を列から"分離(detach)"するために、長さ表示を使用する必要がある。具体的に言うと、も

し復号装置が  $2^{r+q}$  個のワードを記憶し得るメモリを有していると ( $r$  は  $a_k$  のビット長)、復号には次のような手順が必要になる。

- (1) 符号化された列  $C(s a_k)$  の開始点の位置決め。
- (2)  $C(s a_k)$  の次の  $r$  ビット及び保持されている小数部分  $x(s a_k)$  の  $q$  ビットによるメモリのアドレス指定、並びに  $a_k$  及  $l(a_k)$  の検索。
- (3) 復号装置を圧縮された符号列の次の部分と整列させるために、符号列を  $y(s a_k)$  ビットだけ左シフト。

本発明に従う短縮法は、ブロック化が不要だけでなく、有限のメモリしか必要とせず、更にエラーの制御及び符号ワード・テーブルのためのメモリ容量の制御も可能である。

例えば、2進アルファベットのハフマン符号化法においては、エントロピー即ち符号化されたシンボル当りの理論上の最小平均長を  $H(s)$ 、符号化

特開昭53-110338(5)

されたシンボル当りの平均長を  $L_n/n$  としたとき、 $H(s) \leq L_n/n < H(s) + 1/n$  なる不等式(シャノンの第1定理参照)を満足させるためには、 $n$  ビットのブロック化が必要である。ハフマン符号ワードのためのテーブルは、 $2^n$  個の符号ワードを記憶し得る容量を有している必要があり、また多くの符号ワードは、 $n$  よりも多いビット数を有する。

ところが、算術符号化法においては、同じ上限  $H(s) + 1/n$  に対し、大きさが  $n$  のオーダーであるテーブルを用いることができる。この場合の各ワードの長さは、 $\log_2 n$  のオーダーである。

また、算術符号化法は、符号化文字の短縮の他にも、現在の希薄行列(sparse matrix)法とは別の、非符号化データ(ラスタ符号化データ)の列に対する圧縮法をも提供する。ここで重要なことは、符号化データ及び非符号化データに対して別々の手法乃至は機構を用いるものとは異なり、2進列符号化のための単一の短縮法が存在するであろうということである。

以下、図面を参照しながら、本発明の実施例について説明する。

第1A図は、記憶及び伝送システムと共に用いられる2進列符号化装置の概略を示したものである。情報源1は、情報源アルファベット  $\{a_1, a_2, \dots, a_k, \dots, a_N\}$  から選択された一連のシンボル  $a_k$  を出力する。これらのシンボルは、バス3を介してバッファ/コントローラ5へ直列に供給され、そこに累積される。このバッファ5は、情報源1と伝送の前に必要な信号の条件付け又は符号変換との間における必要な速度整合を行なう。記憶/伝送システムは、例えば、利用装置29を駆動する後入れ先出し(LIFO)記憶ユニット15を含んでいる。通常、速度整合及び符号変換は、利用装置の側においても必要であるから、記憶ユニット15と利用装置29との間にバス17及び27を介して別のバッファ/コントローラ19が接続される。

バッファ5に累積された情報源1からのシンボルは、次にバス7を介して符号化装置9の如き変

換装置へ送られる。符号化装置9は1対1の写像(変換)を実行した後、バス11を介して変換されたシンボルをバッファ5へ送る。刻時動作及び各符号文字の長さに関する制御情報は、別の並列バス12を介して送られる。同様に、記憶ユニット15から受取られてバッファ19に累積された符号化文字の変換は、復号装置23で行なわれる。復号装置23は、情報源1から発生された元のシンボルを再生するものであるが、バッファ19からバス21を介して供給された2進列を正しく解析するためには、元の文字及び長さ情報を引出さねばならない。長さ情報はバス22を介して送られ、情報源文字データはバス25を介して送られる。

ここで、第1B図及び第1C図を参照しながら、従来のハフマン法に従う符号化及び復号について説明しておく。ハフマン符号は、各情報源シンボルの事前生起確率  $p(a_k)$  ( $k=1, 2, \dots, N$ ) を利用して、可変長  $l(a_k)$  の対応する符号ワードを割当てたものである。この場合、 $p(a_k)$

及び  $L(a_k)$  については、 $p(a_k) \geq 2^{-L(a_k)}$  及び  $\sum_{k=1}^N 2^{-L(a_k)} \leq 1$  が成立している。符号化列が "0" 及び "1" の組合わせから成つていると、 $L(a_k)$  は、シンボル  $a_k$  に対するハフマン符号ワードのビット数に相当する。

ハフマン符号化法においては、情報源シンボル列が受取られると、各情報源シンボル  $a_k$  に応答してメモリから符号ワード  $A_k$  及び長さ表示  $L(a_k)$  が検索され、次いで、前に符号化された列  $C$  が  $L(a_k)$  ビットだけ右又は左にシフトされた後、符号ワード  $A_k$  が付加される。従つて、情報源シンボル列  $a_i a_j a_k$  をハフマン法で符号化した場合の符号列は、 $A_i A_j A_k$  で表わされる。これは、符号ワードの連結と呼ばれている。

第1B図に示したハフマン符号化装置においては、第1A図のパツファ5からバス7を介して供給された情報源シンボル  $a_i$ 、 $a_j$ 、 $a_k$  は一旦入力レジスタ31に保持される。メモリ・アドレス  $a_k$  によつてアドレス指定されるメモリ33の特定の記憶位置には、対応するハフマン符号ワ

ードを記憶し得るメモリに対してはメモリ47をアドレス指定するために、最初の  $\log_2(1/p_m)$  ビットが入力レジスタ43へ送られる。メモリ47から読出される情報は、情報源シンボル  $a_k$  及びその長さ  $L(a_k)$  である。 $L(a_k)$  はレジスタ49へロードされ、 $a_k$  は出力レジスタ51へロードされる。レジスタ49の内容は、次の符号ワードの復号のために新しい開始点を整列させるようにレジスタ43をシフトさせるため、シフト制御装置45へ送られる。

ハフマン型の圧縮符号ワードの長さは、整数値に限定されることは明らかである。符号化における一般的な原則として、各符号ワードの長さ  $L(a_k)$  と、情報源アルファベットにおける対応するシンボル  $a_k$  の相対生起頻度  $p(a_k)$  とは、次のような関係を有している。

$$p(a_k) \geq 2^{-L(a_k)}$$

これを書き直すと、

$$p(a_k) = 2^{-L(a_k)} + E$$

特開昭53-110338(6)

ド  $A_k$  及び長さ表示  $L(a_k)$  が記憶されている。メモリ33から読出された符号ワード  $A_k$  はレジスタ35へロードされ、長さ表示  $L(a_k)$  はレジスタ37へロードされる。

レジスタ35へロードされた  $A_k$  は、次に出力シフト・レジスタ41の所定の位置へ送られ、一方、レジスタ37へロードされた  $L(a_k)$  は、シフト制御装置39へ送られる。シフト制御装置39は、次の符号ワードを出力レジスタ41へロードし得るようにすると共に転送バス11へ符号ワードを出力するため、符号ワード  $A_k$  を所定の位置数だけシフトさせる。

第1C図のハフマン復号装置は、米国特許第3883847号明細書に示されているような型のものである。パツファ/コントローラ19からの圧縮された符号ワード列は、バス21を介して入力レジスタ43へロードされる。最大符号ワード長は先験的に知られている。従つて、最長の単位符号ワードの長さ(ビット数)を  $\log_2(1/p_m)$  としたときに、 $\log_2(1/p_m)$  個の

となり、従つて次式が得られる。

$$L(a_k) = \log_2 \{ 1/p(a_k) \} + E'$$

これから明らかなように、もし  $L(a_k)$  が整数であれば、 $p(a_k)$  は無理数になる。従つて、符号ワード長が整数値をとる圧縮符号化系の圧縮効率、符号ワード長に小数部分が含まれるような符号化系の効率よりも劣っている。可算符号及び算術列符号には、このような小数部分が含まれる。

#### 本発明に従う符号化アルゴリズム

情報源シンボル  $a_i$  及び  $a_j$  から成るシンボル列  $s = a_i a_j$  を考えると、これに新しいシンボル  $a_k$  を付加した場合には、シンボル列は  $s a_k = a_i a_j a_k$  と表わせる。ここで、シンボル列  $s$  を符号化装置へ供給したときの出力を  $C(s)$  と表わすと、可算符号化及び列符号化においては、 $s \rightarrow C(s)$  及び  $s a_k \rightarrow C(s a_k)$  という対応関係がある。符号化は、次のようにして帰納的に定義さ

れる。

$$L(s a_k) = \text{長さ表示} = L(s) + L(a_k)$$

$$C(s a_k) = \text{圧縮された符号列} = C(s) + P_{k-1} \cdot 2^{L(s a_k)}$$

上式中の  $L(a_k)$  及び  $P_{k-1}$  は次式で表わされる。

$$L(a_k) = L \circ g_2 \{ 1 / p(a_k) \} ,$$

$$P_{k-1} = \sum_{i=1}^{k-1} p(a_i) ; P_0 = 0 ; P_N = 1$$

但し  $L(a_k)$  には

$$\sum_{k=1}^N 2^{-L(a_k)} \leq 1$$

という制限がある。

長さ表示  $L(s a_k)$  を、整数成分及び小数成分に分けて考えると、圧縮された符号列  $C(s a_k)$  の計算をより簡単に行なうことができる。即ち、

$$L(s a_k) = y(s a_k) (\text{整数}) + x(s a_k) (\text{小数})$$

するための算術列符号化装置の一例を第2図に示す。図示の符号化装置は、バス7上の情報源シンボル  $a_k$  に応答して、バス11へ圧縮された符号列  $C(s a_k)$  を供給し、更にバス12へ制御情報を供給する。この制御情報には、線69へ供給される被保持小数成分及び線77へ供給されるシフト量が含まれる。

#### タイミング

第2図の符号化装置及び第3図の復号装置のためのタイミング方式は、同期式単相型の簡単なもので、回路遅延、フリップフロップの設定に要する時間及び伝播遅延が最悪の場合にもうまく動作させるため、刻時パルス間のパルス間隔は十分に長くとられている。これについては、1972年にマグロー・ヒル社から発行されたジェイ・ビー・ビートマン著の「デジタル・システムの設計 (The Design of Digital Systems)」の161乃至170頁を参照されたい。第4A図は、符号化装置における最悪の場合の各遅延の様

$$C(s a_k) = C(s) + P_{k-1} \cdot 2^{y(s a_k)} \cdot 2^{x(s a_k)}$$

ここで、 $A_k = P_{k-1} \cdot 2^{x(s a_k)}$  とおくと、

$$C(s a_k) = C(s) + A_k \cdot 2^{y(s a_k)}$$

これは次のように書き直せる。

$$2^{-y(s a_k)} \cdot C(s a_k) = 2^{-y(s a_k)} \cdot C(s) + A_k$$

上式中の  $A_k$  については、 $0 \leq P_{k-1} < 1$  及び  $0 \leq x(s a_k) < 1$  であるから、 $0 \leq A_k < 2$  が成立する。また、 $A_k$  は  $k$  及び  $x$  の関数であるから、その値は、 $k$  及び  $x$  に基づくテーブル索引を利用して得ることができる。しかしながら、符号列  $C(s a_k)$  の生成においては、先行符号列  $C(s)$  とテーブルから引出された値  $A_k$  とを正しく整列させることが必要である。言い換えれば、先行符号列とテーブルからの値との間の相対的な位置関係を維持しておくことが重要である。

上述の2種類の符号化手順のうちの後者を実施

子を刻時パルス  $CP$  と共に示したもので、第4B図は同じく復号装置におけるものである。刻時スキューは、最大許容公差の範囲内にあるものとする(上述のテキストの167乃至170頁参照)。

第2図に示した符号化用メモリ57及び59並びに第3図に示した復号用メモリ99、109及び117には、初期設定段階の間に必要な値がロードされる。これらのメモリは、読出し/書込みメモリ、読出し専用メモリ及びプログラム可能な読出し専用メモリの何れであつてもよい。このようなメモリをロードするための手段については周知であるから、ここでは特に述べない。Cレジスタ87は、前に符号化された列の上位  $r$  ビット即ち算術オペランド部分を保持し、Xレジスタ67は、 $q$  ビットの被保持小数成分を保持する。これらのレジスタは、普通「0」に初期設定されるが、Xレジスタ67には、任意の値を初期設定してもよい。Cレジスタ87の初期設定値は、あとで説明する値  $\max C_x$  以下になるように制限される。可算符号化においては、先行符号列  $C(s)$  の算術オ

ベラント部分は、この列C(a)の全体であるが、本発明では、Cレジスタ87は、先行符号列C(a)の算術オペラントを構成し得る上位のrビットのみから成る被保持列セグメントを保持する。残りの下位ビットはすべてシフトされるだけである。Cレジスタ87の内容は、算術的使用の前に、 $y(a_k)$ ビットだけ右方向にシフトされる。この結果、Cレジスタ87の右端の $y(a_k)$ ビット(出力列セグメントと呼ばれる)は、バス11を介して第1A図に示したバッファ/コントローラ5へ送られ、このとき、シフト量 $y(a_k)$ を表わす制御信号もバス12を介して送られる。この制御信号は、バッファ/コントローラ5に受取られるビットの数を表わす。

各情報源シンボル $a_k$ は、入力レジスタ31に受取られた後、変換器53へ送られる。変換器53は、シンボル $a_k$ をその位置番号に対応する数値 $k$ (nビット)へ変換する。この数値 $k$ は、バス55を介してメモリ57へ供給される。メモリ57は、 $L(a_k) = \log_2 \{1/p(a_k)\} = y(a_k) + x(a_k)$

7にロードされている。 $x(a_k)$ は、テーブル索引によつてメモリ59から被加数 $A_k$ をアクセスするために、 $k$ と共に用いられる。メモリ59から取出されたrビットの $A_k$ は、バス89を介して加算器83へ送られる。新しい符号化列C( $a_k$ )は、 $A_k$ と右シフト装置79で適切にシフトされた後のC(a)とを加算器83で加算することによつて得られる。ここで注意すべきは、先行符号化列C(a)のrビットの被保持列セグメントのうちの低位の $y(a_k)$ ビットがバッファ/コントローラ5へ送られるということである。次の刻時サイクルにおいては、符号化列C( $a_k$ )の新しい被保持セグメントが、バス85を介してCレジスタ87へ送られる。連続する刻時パルスCP間の時間間隔、関連する回路の最大遅延を考慮して、 $x(a_k)$ 及びC( $a_k$ )が安定化し得るように即ち定常状態をとり得るように、設定される。各レジスタは、1973年にテキサス・インスツルメンツ社から発行された“設計技術者のためのTTLデータ・ブック(The TTL

特開昭53-110338(8)

で表わされる長さ $L(a_k)$ を含んでいる。ここで、整数成分 $y(a_k)$ はiビットであり、小数成分 $x(a_k)$ はqビットである。

第2図及び第3図に示した添字付きの短い斜線は、各々のバスが添字の数だけの平行導体から成っていることを表わしている。例えば、バス55はn本の導体を含み、またバス61及び71は各々q本及びi本の導体を含んでいる。また、丸印で囲んだ大文字のアルファベットは、同じアルファベット同志が接続されることを表わしている。

整数値 $y(a_k)$ は、加算器63及び+1加算器75から得られる。小数値 $x(a_k)$ は、加算器63及びXレジスタ67から得られる。 $y(a_k)$ の値は、先行符号列C(a)とバス89上の $A_k$ とを並列させて、これらを加算器83で加算することによつてC( $a_k$ )の上位のrデジットを得る際に、右シフト装置79でシフトされるC(a)のデジットの数を表わす。

小数値 $x(a_k)$ は次の刻時サイクルの開始時にXレジスタ67へロードされることになるが、現サイクルの開始時点においては、前的小数値 $x(a)$ がXレジスタ6

Data Book for Design Engineers)の363乃至368に記載されている74TTLファミリー、型番SN74174のような“D”型のフリップフロップで構成することができる。これに関して、バッファ/コントローラ5は、刻時パルスの発生毎に符号化装置9へ新しいシンボル $a_k$ を供給すると共に、シフト動作によつて出力された先行符号化列C(a)の被保持列セグメントの低位のyデジットを取り去るよう構成されてもよい。最後のシンボルの符号化が終ると、バッファ/コントローラ5はC( $a_k$ )の最終内容及び最終被保持小数成分 $x(a_k)$ を受取る。

#### パラメータ“r”及び“q”の大きさ

前述のように、C( $a_k$ )は、符号化装置9に供給されたシンボル $a_k$ に回答して符号化された列を表わし、C(a)は先行符号列を表わしている。符号列の長さ(ビット数)は、前に符号化された文字のすべての長さ表示 $L(a_k)$ の和になつている。しかしながら、符号化装置9の内部に保持



される符号列  $C(s)$  のビットは、上位の  $r$  ビットだけである。新しい長さ表示  $L(s, a_k)$  が得られたときには、 $C(s)$  の  $r$  個の被保持ビットのうちの下位の  $y(s, a_k)$  ビットがシフト動作によつて出力される。この結果、 $C(s)$  の  $r - y(s, a_k)$  ビットが算術オペランドとして残され、 $A_k$  の下位ビットと整列される。 $L(s, a_k) = L(s) + L(a_k) = y(s, a_k) + x(s, a_k)$  であることを想起されたい。しかしながら、 $L(s)$  のうち保持されている部分は小数成分  $x(s)$  だけである。整数成分  $y(s)$  は、前の符号化処理で得られた符号列  $C(s)$  の右シフト量に対応している。従つて、前述のように、 $L(s, a_k)$  は  $x(s) + L(a_k) =$  整数  $y(s, a_k)$  + 小数  $x(s, a_k)$  で表わされる。

既に述べたように、符号化操作においては、 $A_k$  の値はテーブル索引によつてメモリ 59 から得られる。メモリ 59 をアクセスするためのアドレス・ビットは  $a_k$  の位置番号  $k$  を表わす  $n$  ビット及び被保持小数成分  $x(s, a_k)$  の  $q$  ビットから成

っている。この場合、 $q$  の値は、独立設計パラメータとして扱われる。

メモリ 59 から取出された  $r$  ビットの  $A_k$  は、 $C(s)$  の  $r - y(s, a_k)$  ビットと正確に整列されねばならない。これを達成するため、 $C(s)$  は  $y(s, a_k)$  ビット位置だけ右方向にシフトされる。これにより、 $A_k$  及びシフトされた  $C(s)$  の最下位ビットが正しく整列する。 $A_k$  が左シフトされた場合には、 $C(s, a_k) = C(s) + A_k \cdot 2^{y(s, a_k)}$  となり、もし  $C(s)$  が  $A_k$  を基準にして右シフトされると、これは上式の両辺に  $2^{-y(s, a_k)}$  を乗じたことになる。即ち、

$$2^{-y(s, a_k)} C(s, a_k) = 2^{-y(s, a_k)} C(s) + A_k$$

となる。これが(4)式のように書き直せることは前述の通りである。

$r$  の値は、 $A_k$  に要求される精度を決定する。設計上の重要な要素の1つに、生起頻度の最も低い情報源シンボルの生起確率  $p_m$  がある。 $A$  は累積確率  $P$  と  $2^x$  との積で表わされるから、

$$A_k = P_{k-1} 2^{x(s, a_k)}$$

$$A_{k+1} = P_k 2^{x(s', a_{k+1})}$$

において、 $x(s, a_k) = x(s', a_{k+1})$  となる可能性を考えると、 $A$  同志を区別するためには、 $P$  同志を区別することが必要になる。従つて、 $A_k$  のビット数  $r$  は、 $P_k > P_{k-1}$  を満足させるため  $r \geq \log(1/p_m)$  でなければならない。 $r$  を決定する際に考慮すべき他の事柄は、あとの説明から明らかとなろう。被保持小数成分  $x(s, a_k)$  のビット数  $q$  は、実際の符号ワード長がどれ程理想に近いかを知るための基準になる。理想的な符号ワード長が  $\log_2 \{1/p(a_k)\}$  であることは明らかであろう。ハフマン符号ワードの長さは整数値に限定されるので、これを上述の理想ワード長に近づけることはできない。 $q$  の値を大きくすればする程、符号ワードの長さは理想ワード長に近づく。

#### 右シフト対左シフト

まず、情報源シンボル列におけるシンボルの位置を添字で表わして、 $C_0 = 0$  を符号列の初期値とし、 $L_0 = 0$  を長さの初期値とする。シンボル列における最初のシンボル  $a_1$  を符号化するためには、 $L_1 = L_1 = y_1 + x_1$  及び  $A_1$  を決定する必要がある。 $A_1$  を  $y_1$  ビットだけ左シフトして、これを  $C_0$  に加算する代りに、 $C_0$  を  $y_1$  ビットだけ右シフトして、 $A_1$  に加算することも可能である。即ち、

$$C_1 = A_1 + C_0 \cdot 2^{-y_1}$$

となる。

次に、シンボル列  $a_1 a_2$  を符号化する場合に、新しい相対長さ表示は、 $L_2 = L_2 + x_1$  である。このステップでは、小数成分  $x_1$  のビットはシフトされずに  $L_2$  に加算される。 $L_2$  の整数成分を  $y_2$  で表わし、小数成分を  $x_2$  で表わすと、新しい符号列  $C_2$  は次のようになる。

$$C_2 = C_1 \cdot 2^{-y_2} + A_2$$

符号列  $C_{i-1}$  及び被保持小数成分  $x_{i-1}$  から  $i$  番目の情報源シンボル  $a_i$  を符号化する場合の式は、

$$L_i = L_{i-1} + x_{i-1} = y_i + x_i$$

$$C_i = A_i + 2^{-y_i} C_{i-1}$$

となる。被保持小数成分  $x(a)$  を有する符号列  $C(a)$  として既に符号化されている情報源シンボル列  $s$  へ情報源シンボル  $a_k$  を付加した場合について上式を書き直すと次のようになる。

$$L(sa_k) = L(a_k) + x(a) = y(sa_k) + x(sa_k)$$

$$C(sa_k) = A_k + C(a) \cdot 2^{-y(sa_k)}$$

$A_k$  及び " $r$ " についての要件

ここで、復号を可能にする上で重要な事柄について説明する。各シンボル  $a_k$  には、符号化時の  $C(a)$  のシフト量に影響を及ぼす長さ  $L(a_k)$  が対応している。このシフト量に影響を及ぼす他の

要素は、前の被保持小数成分  $x(a)$  である。シフト量  $y(sa_k)$ 、新しい被保持小数成分  $x(sa_k)$  及び被加数  $A_k$  は、算術符号化不等式と呼ばれる次の不等式を満たす。

$$A_{k+1} > A_k + 2^{-y(sa_k)} \cdot C(a)$$

言い換えれば、隣り合うシンボル  $a_k$  及び  $a_{k+1}$  に各々対応する被加数  $A_k$  及び  $A_{k+1}$  は、適切な大きさだけ異なっていなければならず、更にシフト量は、 $2^{-y(sa_k)} \cdot C(a)$  を十分に小さくして、和が  $A_{k+1}$  よりも小さくなるように、十分大きくなければならない。もし上述の条件が満たされなければ、算術符号化を首尾よく行なうことはできない。 $C(sa_k)$  を復号する際には、 $C(sa_k)$  は  $A_k$  以上であることがわかつている ( $A_k$  は被加数だから)。また、上述の不等式から、 $C(sa_k)$  が  $A_{k+1}$  よりも小さいこともわかつており、更に  $C(sa_k)$  のオペランド部分が  $r$  ビットであること及び  $x(sa_k)$  が  $q$  ビットであることもわかつている。従つて、 $C$

$sa_k$ ) 及び  $x(sa_k)$  の可能性のある組合せの数は  $2^{r+q}$  である。各々の組合せについて、 $C(sa_k)$  を生ぜしめたシンボル  $a_k$  が予め決められ、そしてその序数即ち位置番号  $k$  が  $2^{r+q}$  ワードのメモリの対応するワード位置に記憶される。

次に、符号化及び復号を可能にする  $A_k$  のテーブルについて考える。上述の算術符号化不等式から、

$$A_{k+1} - A_k > C(a) \cdot 2^{-y(sa_k)}$$

であることがわかる。

実際には、 $A_k$  の値は  $r$  ビットでの近似値である。被加数  $A_k$  の精度は、 $C(a)$  の達成可能な最大値によつて左右される。これは、先行の被保持小数成分の値  $x'$  及び前に符号化されたシンボルのインデックス  $k'$  によつて異なる。現在の、被保持小数値を  $x$  で表わし、前の値を  $x'$  で表わすと、 $y(sa_k) + x = L(a_k) + x'$  であるから、 $y(sa_k) = L(a_k) + x' - x$  となる。こ

れから、 $x$  及び  $k$  の与えられた値について、 $x'$  が  $(1 + x - L(a_k))$  の小数成分の小数成分に等しいことがわかる。 $A_N(x) > A_{N-1}(x) > \dots > A_1(x)$  であるから、与えられた被保持小数値  $x$  に対して、 $\max C_x > A_N(x)$  である。ここで、 $\max C_x$  は、現在の被保持小数値  $x$  を用いて生成 (符号化) され得る  $C(sa_k)$  の最大値を表わしている。従つて、復号の基準は、 $L(a_k)$  を  $L_k$  で表わすと、 $k < N$  なる各  $k$  及び可能な小数値  $x$  に対して、

$$A_{k+1}(x) - A_k(x) > \max C_x \cdot 2^{L_k - x' + x}$$

である。この不等式は、 $A_N(x)$  について直接の制限を加えるものではない。しかしながら、もし  $A_N(x')$  が大き過ぎると、 $\max C_x$  も大きくなつて、 $A_{k+1}(x) - A_k(x)$  を大きくし、更にこれにより  $A_N(x)$  を大きくする。この問題を解決するためには、 $\max C_x$  及び  $A_N(x)$  の上限を定めなければならない。すべての  $x$  について、もし  $A_N(x) \cdot 2^{-x} + 2^{-L_N} \leq 1$  であれば、即ち、 $A_N$

$(x) + 2^{x-LN} \leq 2^x$  であれば、 $\max C_x \leq 2$  が成立する。

C が最大であつて、シンボル  $a_N$  が符号化される場合には、

$$\max C_x = A_N(x) + \max C_{x'} \cdot 2^{-LN-x'} + x$$

となり、これの両辺に  $2^{-x}$  を乗じると次式が得られる。

$$\max C_x \cdot 2^{-x} = A_N(x) \cdot 2^{-x} + \max C_{x'} \cdot 2^{-x'} \cdot 2^{-LN}$$

上式から、すべての  $x$  について、もし  $\max C_x = 2^x$  であれば、 $A_N(x) \cdot 2^{-x} + 2^{-LN} = 1$  となる。逆も同様である。また、すべての  $x$  について、もし  $A_N(x) \cdot 2^{-x} + 2^{-LN} < 1$  であれば、 $\max C_x < 2^x$  が成立することも導ける。 $A_k(x)$  の値の決定については、あとの実例のところで説明する。

#### 復号アルゴリズム

第3図に示した算術復号装置はバッファ/コ

れた最後の符号列  $C(a_k)$  を受取るだけでよく、先行の符号列を受取つて保持しておく必要はない。先行の符号列、例えば  $C(a)$  は、次のようにして得られる。まず、バス97を介して得られる符号列  $C(a_k)$  と、メモリ109からバス111へ脱出された  $A_k$  との差が、減算器113で計算され、この結果、 $y(a_k)$  ビットだけ右シフトされた  $C(a)$  (即ち  $2^{-y(a_k)} \cdot C(a)$ ) がバス115へ出力される。バス115上のシフトされた  $C(a)$  は、符号化されたときの形へ戻すために、左シフト装置91へ供給される。このとき、バッファ/コントローラ19は、左シフト装置91へ“充填”ビット即ち符号化のときにバッファ/コントローラ5へシフトされた下位の  $y(a_k)$  ビットを供給する。左シフト装置91は、バス123から入力される  $y(a_k)$  に応答して、 $y(a_k)$  ビットだけ左シフトを行ない、これによりバス93上に  $C(a)$  が得られる。

$L(a_k)$  の整数成分  $y(a_k)$  及び小数成分  $x(a_k)$  は、バス107を介してメモリ117

特開昭53-110338(11)

ントローラ19からバス21を介して圧縮された符号化列を受取り、そして復号されたシンボルをバス25を介してバッファ/コントローラ19へ送る。適切な圧縮されたビット列セグメントをシフトさせるのに必要な長さ情報は、バス22の一部である線123を介して送られる。

復号装置を初期設定するため、最後の被保持小数値  $x(a_k)$  がバッファ/コントローラ19からバス22の線101を介してXレジスタ103へロードされる。 $C(a_k)$  の上位の  $r$  ビットは、Cレジスタ95に置かれる。Cレジスタ95からの  $r$  ビットは、Xレジスタ103からの  $q$  ビットと共に、復号されるべき最終シンボル  $a_k$  の位置番号  $k$  をメモリ99から得るためのアドレスを構成する。 $A_k$  の値は、メモリ99から線107へ取出された  $k$  を表わす  $n$  ビット及び線105を介して供給される  $x(a_k)$  を表わす  $q$  ビットでメモリ109をアクセスすることにより得られる。

符号化された要素を抽出するためには、圧縮さ

をアクセスすることによつて得られる。図示のように、 $y(a_k)$  はバス119を介して+1加算器121へ入力され、 $x(a_k)$  は減算器125の減数入力へ供給される。減算器125の被減数入力には、小数成分  $x(a_k)$  が供給される。減算器125は、 $x(a_k) - x(a_k)$  を計算して、バス129へ小数成分  $x(a)$  を出力し、更に、もしあれば反転器127へ借り信号を送る。新しい符号ワード  $C(a_k)$  の長さ表示  $L(a_k)$  の整数成分  $y(a_k)$  は、反転器127からの借り信号の補数と  $y(a_k)$  とを加算器121で加算することによつて得られる(+1加算)。

最終シンボルの復号時には、レジスタ95は符号列Cの初期値(普通は0)を、そしてレジスタ103は被保持小数成分の初期値を各々含んでいるはずである。これは、符号列の記憶及び伝送の検査に利用できる。

$a_k$  についての実際のシンボル値は、 $k$  をバス105から変換器131へ供給することによつて得られる。この変換操作は、第2図に示した符号

装置における変換器 53 の操作とは逆になつてゐる。変換器 131 の出力は、バス 25 を介してバッファ/コントローラ 19 へ送られる。

最後に、これまで説明してきた算術符号化法を実際に適用した場合について述べる。

#### 実例

##### 1. 被加数 $A_k$ の決定

情報源アルファベット  $S$  は 4 つのシンボル  $a_1$ 、 $a_2$ 、 $a_3$  及び  $a_4$  から成っており、各々の生起確率を  $p_1 = 0.65$ 、 $p_2 = 0.075$ 、 $p_3 = 0.025$ 、 $p_4 = 0.25$  とする。このような情報源  $S$  においては、エントロピー  $H$  は、1 シンボル当り 1.31729 ビットであり、対応するハフマン符号の長さは、 $\ell_1 = 1$ 、 $\ell_2 = 3$  及び  $\ell_4 = 2$  である。シンボル当りの平均長は 1.45 ビットであり、これは、情報源  $S$  のシンボルを 2 ビットとすると、即ち、 $a_1 = 00$ 、 $a_2 = 01$ 、 $a_3 = 10$  及び  $a_4 = 11$  とすると、平均長が情報源シンボルの長さの 72.5% になる符号列を与える。

ては、一時に 2 個のシンボルをブロック化して、情報源符号アルファベットの要素を 16 個に増やすことが必要である。これを行なうと、圧縮率は 66.76% になる。

$\ell_1 = 0.5$ 、 $\ell_2 = 4$ 、 $\ell_3 = 4.5$  及び  $\ell_4 = 2.5$  について  $A_k$  の値を求める場合には、これらの長さを与える理想確率  $p'_k = 2^{-\ell_k}$  を考えればよい。即ち、 $p'_1 = 0.7071$ 、 $p'_2 = 0.0625$ 、 $p'_3 = 0.04419$  及び  $p'_4 = 0.17678$  である。

小数成分  $x$  が 0 の場合には、 $A_k(0)$  の値は、理想累積確率  $P'_{k-1}$  の近似値 (小数ビットは  $q$  個) を要する。最小の確率  $p'_3$  を 2 進表示すると、0.0000101101... であるから、少なくとも 5 個の小数ビットが必要である。

$$\begin{aligned} A'_1(0) &= P'_0 = 0 &= 0.00\dots \\ A'_2(0) &= P'_1 = 0.7071067\dots = 0.10110101000\dots \\ A'_3(0) &= P'_2 = 0.7696067\dots = 0.11000101000\dots \\ A'_4(0) &= P'_3 = 0.8138009\dots = 0.11010000010\dots \end{aligned}$$

特開昭53-110338(12)

理論上の最良の圧縮率は約 65.86% 即ち 1.31729 ビットである。

長さ表示に 1 つの小数ビットを用いる ( $q = 1$ ) 算術符号化を考える。ここでの問題は、1 つの 2 進小数ビットを有する有理数であつて、クラフトの不等式を満たしながら平均長を最小にする一組の長さ  $\ell_1$ 、 $\ell_2$ 、 $\ell_3$  及び  $\ell_4$  を見つけることである ( $\ell_k \times 2^1$  は整数になる)。この場合、 $\log_2 (1/p_i)$  から理想長を計算すると、0.62149、3.73697、5.32193 及び 2 となる。エントロピーは、これらの理想長から計算することができる。これらの値から長さを求めると、 $\ell_1 = 0.5$ 、 $\ell_2 = 4$ 、 $\ell_3 = 4.5$ 、及び  $\ell_4 = 2.5$  となり、これらをクラフトの不等式の左辺に代入すると 0.990578 となつて、1 よりも小さい。これらの長さから計算される平均長は 1.3625 であつて、68.125% の圧縮率が得られ、これは 72.5% の圧縮率を与える上述の単一シンボル・ハフマン符号化よりも優れている。圧縮率を改善するため、ハフマン符号におい

次になすべきことは、復号可能性の基準を満たすように、 $A'_k(0)$  を  $r$  ビット (1 整数ビット及び  $r-1$  個の小数ビット) で近似することである。 $\max C_0 < 2^0$  及び  $\max C_{0.5} < 2^{0.5}$  が満足されているものとする。 $r-1 (= 5)$  個の小数ビットに丸めると次のようになる。

第1試行 ( $x=0$ )	$A_{k+1}-A_k$	$2^{x'} \cdot 2^{-L_k - x' + x}$
$A_1(0)=0.0$	0.71825	0.5
$A_2(0)=(0.10111)_2=0.71875$	0.0625	0
$A_3(0)=(0.11001)_2=0.78125$	0.0625	0.5
$A_4(0)=(0.11011)_2=0.84375$		0.5

第2試行 ( $x=0$ )	$A_{k+1}-A_k$	$2^{x'} \cdot 2^{-L_k - x' + x}$
$A_1(0)=0.0$	0.703125	0.7071067812
$A_2(0)=(0.101101)_2=0.703125$	0.0625	0.0625
$A_3(0)=(0.110001)_2=0.75625$	0.0625	0.044194174
$A_4(0)=(0.110101)_2=0.828125$		0.1767766953

特開昭53-110338(13)

$\max C_0 = 1$  及び  $\max C_{0.5} = 1.1414$  とすると、 $k=1, 2$  及び  $3$  に対しては、 $A_{k+1}(0) - A_k(0) > \max C_x$ ,  $2^{L_k + x'}$  が成立する。 $k=4$  に対しては、 $A_4(0) - 2^{-L_4} = 1.020526695 > 1$  となるが、これは、小数ビットの数 ( $r-1$ ) を  $6$  にして、第1試行における  $A_2(0)$ 、 $A_3(0)$  及び  $A_4(0)$  から  $2^{-6}$  を引くことによつて補正することができる。

この場合、 $A_4(0) - A_3(0) = 0.0625$  となつて、 $2^{-L_3} = 0.044194174$  との差が  $2^{-6}$  よりも大きく、従つて  $A_4$  を  $2^{-6}$  だけ小さくすることができる。

第1試行 ( $x=0.5$ )	$A_{k+1}-A_k$	$x'$	$2^{x'} \cdot 2^{-L_k} - x' + x$	第3試行 ( $x=0$ )	$A_{k+1}-A_k$	$2^{x'} \cdot 2^{-L_k} - x' + x$
$A_1(0.5)=0.0000000=0$	1	0	1	$A_1(0)=0.0$	0.703125	0.7071067812
$A_2(0.5)=1.0000000=1$	0.9375	0.5	0.08838834765	$A_2(0)=(0.101101)_2=0.703125$	0.0625	0.0625
$A_3(0.5)=1.000110=1.09375$	0.0625	0	0.0625	$A_3(0)=(0.110001)_2=0.765625$	0.046875	0.044194174
$A_4(0.5)=1.001010=1.15625$	0	0	0.25	$A_4(0)=(0.110100)_2=0.8125$	0.1767766953	

この場合は、 $A_4 + 2^{-L_4} = 0.8125 + 2^{-2.5}$   
 $= 0.9892766953 < 2^0 = 1$ となるが、  
 $A_2(0) - A_1(0)$ が $2^{-L_1}$ よりも小さくなるという  
 問題がある。しかしながら、もし $\max C_{0.5}$ が  
 十分に小さければ、復号可能性の基準は、 $\max$   
 $C_0 (< 2^0)$ 及び $\max C_{0.5} (< 2^{0.5})$ の  
 実際の値に基づいて満たされ得る。

$x=0$ の場合と同じようにして、 $x=0.5$ の  
 ときの理想被加数 $A'_k(0.5) = 2^{0.5} \cdot p'_{k-1}$   
 を求めてみる。

$$\begin{aligned} A'_1(0.5) &= 0.0 & = 0 \\ A'_2(0.5) &= 1 & = (10000000)_2 \\ A'_3(0.5) &= 1.088388348 = 1.000101110 + \\ A'_4(0.5) &= 1.150888348 = 1.001001110 + \end{aligned}$$

上述の理想値を丸めると次のようになる。

$A_4(0.5)$ を調べてみると、 $A_4(0.5) + 2^{0.5-L_4}$   
 $= 1.40625$ であるから、 $2^{0.5} = 1.4142$   
 よりも小さい。 $\max C_0$ 及び $\max C_{0.5}$ は、  
 シンボル $a_4$ の列を符号化することによって決定  
 できる。 $C$ は0に初期設定され、 $C_0$ は $2^0$ だけ  
 大きくされ $C_{0.5}$ は $2^{0.5}$ だけ大きくされる。こ  
 れらの単調増加シーケンスは $\max C_0$ 及び $\max$   
 $C_{0.5}$ に各々収束する。 $\max C_0$ は0.98790  
 32758(2進表示では、0.111111100  
 11100.111001111...001111...  
 )に決定された。 $\max C_{0.5}$ は1.403225  
 806(2進表示では1.0111001111001  
 11001111...)である。 $\max C_{0.5}$ のこ  
 の新しい値を用いると、 $A_2(0) - A_1(0) = 0.70$   
 $3125$ と $\max C_{0.5} \cdot 2^{-L_1 - 0.5} = 1.4$   
 $03225806 \times 0.5 = 0.701612903$   
 との比較から、復号可能性の基準が満たされてい  
 ることがわかる。

もし復号可能性の基準が満たされていないければ、  
 理想被加数 $A'_k(x)$ をより高い精度で近似するよ

うに、小数ビットの数 ( $r-1$ ) を増やす必要がある。  $\varepsilon 2^{-\ell_k} < 1$  である限り、 $q > 0$  (長さ  $\ell_k$  が非整数即ち小数) についての解は常に存在する。

符号化装置 9 及び復号装置 23 のメモリはロードされねばならない。符号化用メモリ 5.7 の内容は次の通りである。なお、添字の "2" は 2 進表示を表わしている。

アドレス(k)	$\ell_k$
00	(0001) <sub>2</sub>
01	(1000) <sub>2</sub>
10	(1001) <sub>2</sub>
11	(0101) <sub>2</sub>

メモリ 5.9 の内容は次の通りである。

アドレス (x, k)	$A_k(x)$
0, 00	(0000000) <sub>2</sub>
0, 01	(0101101) <sub>2</sub>
0, 10	(0110001) <sub>2</sub>
0, 11	(0110100) <sub>2</sub>
1, 00	(0000000) <sub>2</sub>
1, 01	(1000000) <sub>2</sub>
1, 10	(1000110) <sub>2</sub>
1, 11	(1001010) <sub>2</sub>

復号装置 23 においては、メモリ 10.9 及び 11.7 の内容は、符号化装置 9 のメモリ 5.9 及び 5.7 の内容と各々同じであるが、メモリ 9.9 は次のようになっている。

アドレス (x, C <sub>x</sub> )	k についての符号化	シンボル
0, 0000000 ↓	00 ↓	a <sub>1</sub>
0, 0101100 ↓	00 ↓	
0, 0101101 ↓	01 ↓	a <sub>2</sub>
0, 0110000 ↓	01 ↓	
0, 0110001 ↓	10 ↓	a <sub>3</sub>
0, 0110011 ↓	10 ↓	
0, 0110100 ↓	11 ↓	a <sub>4</sub>
0, 1111111 ↓	11 ↓	
1, 0000000 ↓	00 ↓	a <sub>1</sub>
1, 0111111 ↓	00 ↓	
1, 1000000 ↓	01 ↓	a <sub>2</sub>
1, 1000101 ↓	01 ↓	
1, 1000110 ↓	10 ↓	a <sub>3</sub>
1, 1001001 ↓	10 ↓	
1, 1001010 ↓	11 ↓	a <sub>4</sub>
1, 1111111 ↓	11 ↓	

このメモリは、C<sub>x</sub> (及び被保持小数値 x) の値と A<sub>k</sub>(x) との同時比較を可能にする。

## 2. シンボル列の符号化

シンボル列 a<sub>4</sub> a<sub>2</sub> a<sub>1</sub> a<sub>3</sub> が符号化されるものとする。C レジスタ 8.7 の内容を C で表わし、i 番目の符号化ステップにおける符号列の全体を C<sup>(i)</sup> で表わす。符号化ステップは次の通りである。

### 刻時 1

$$a_4 \rightarrow \ell_4 = 2.5, x = 0.5, C^{(1)} = A_4(0.5) = (1001010)_2$$

### 刻時 2

$$(a) a_2 \rightarrow \ell_2 = 4, y = 4, x = 0.5, A_2(0.5) = (1000000)_2$$

$$(b) C^{(1)} \text{ が 4 ビットだけ右シフトされて、} A_2(0.5) \text{ に加算される: } C^{(2)} = 10001001010. \text{ 下位の 4 ビットは、バッファ/コンローラ 5 へ送られる: } C = 10001000.$$

### 刻時 3

- (a)  $a_1 \rightarrow \ell_1 = 0.5$ 、 $y = 1$ 、 $x = 0$ 、 $A_1(0) = 0$
- (b) C は 1 ビットだけ右シフトされて (最下位ビットはバッファ/コントローラ 5 へ送られる)、 $A_1(0)$  に加算される:  $C^{(3)} = 0.10001001010$ 、 $C = 0.100010$ 。

### 刻時 4

- (a)  $a_3 \rightarrow \ell_3 = 4.5$ 、 $y = 4$ 、 $x = 0.5$ 、 $A_3(0.5) = 1.000110$
- (b)  $C^{(3)}$  は 4 ビットだけ右シフトされて (下位の 4 ビットはバッファ/コントローラ 5 へ送られる)、 $A_3(0.5)$  に加算される:  $C^{(4)} = 1.001000001001010$ 、 $C = 1.001000$ 。

### 3. 復号

この場合の符号列は  $C = 1.001000001$

### 刻時 2

- (a) メモリ 99 へアドレス ( $x_1$ 、 $C$ ) = (0、0.100010) を供給して、 $k(00) = a_1$  を取出す。
- (b)  $\ell_1 = 0.5$ 、新  $x = 0.5$ 、 $y = 0.5 + 0.5 = 1$
- (c)  $C = 0.100010$  から  $A_1(0) = 0.000000$  が引かれる。
- (d) 減算結果が 1 ビットだけ左シフトされ、バッファ 19 から 1 つの 0 ビットが供給される:  $C = 1.000100$ 、バッファ内の  $C = 1010$

### 刻時 3

- (a) メモリ 99 へアドレス ( $x$ 、 $C$ ) = (1、1.000100) を供給して、 $k(01) = a_2$  を取出す。
- (b)  $\ell_2 = 4$ 、新  $x = 0.5$ 、 $y = 4.5 - 0.5 = 4$
- (c)  $C = 1.000100$  から  $A_2(0.5) = 1.000000$  が引かれる  $\rightarrow 0.000100$
- (d) 減算結果が 4 ビットだけ左シフトされ、空い

特開昭53-110338(16)

001010 であり、被保持小数成分は  $x = 0.5$  である。復号装置 23 の X レジスタ 103 は  $x = 0$  に初期設定され、7 ビットの C レジスタ 95 には 1.001000 が与えられる。

### 刻時 1

- (a) メモリ 99 へアドレス ( $x$ 、 $C$ ) = (1、1.001000) を供給して、 $k00 = a_3$  を取出す。
- (b)  $\ell_3 = 4.5$ 、新しい  $x = 0$ 、 $y = \ell_3 + \text{新 } x$ 、被保持小数値 = 4
- (c) 1.001000 から  $A_3(0.5) = 1.000110$  が引かれる  $\rightarrow 0.000010$
- (d) この減算結果が 4 ビットだけ左シフトされ ( $\rightarrow 0.10$ )、バッファ/コントローラ 19 からの次の 4 ビット (0010) が C レジスタ 95 に記憶される: 0.100010。(バッファ内の  $C = 01010$ )

た所にバッファ/コントローラ 19 からの 4 ビット (1010) が供給される:  $C = 1.001010$ 、バッファ/コントローラ 19 は空

刻時 4 (バッファ/コントローラ 19 は空であつて、最終ステップであることを知らせる)

- (a) メモリ 99 へアドレス ( $x$ 、 $C$ ) = (1、1.001010) を供給して、 $k00 = a_4$  を取出す。
- (b)  $\ell_4 = 2.5$ 、新  $x = 0$ 、 $y = 2$
- (c)  $C = 1.001010$  から  $A_4(0.5) = 1.001010$  が引かれる  $\rightarrow 0.000000$
- (d) バッファ/コントローラ 19 はシフト量を無視する。復号完了。

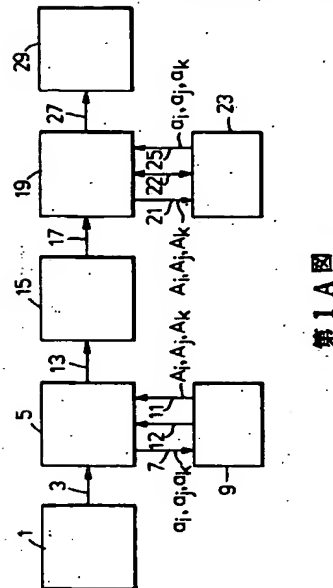
復号が完了すると、C レジスタ 95 及び X レジスタ 103 の内容は 0 になるが、もし異なつた値が含まれていると、何らかのエラーが生じたことになる。



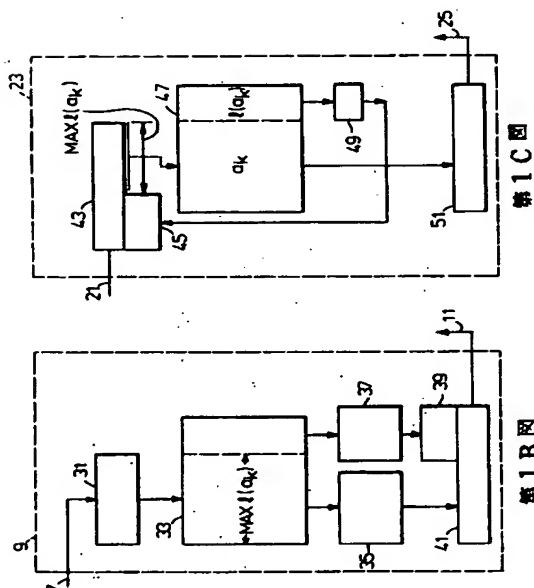
4. 図面の簡単な説明

第1A図は符号化及び復号を行なうシステムの概略を示すブロック図、第1B図はハフマン符号化装置のブロック図、第1C図はハフマン復号装置のブロック図、第2図は本発明に従う符号化装置のブロック図、第3図は第2図の符号化装置に関連して使用される復号装置のブロック図、第4A図は第2図の符号化装置のためのタイミング図、第4B図は第3図の復号装置のためのタイミング図である。

1...情報源、5、19...バッファ/コントローラ、9...符号化装置、15...LIFO記憶ユニット、23...復号装置、29...利用装置、31...入力レジスタ、53...変換器、57、59...メモリ、63、65...加算器、67...Xレジスタ、75...+1加算器、79...右シフト装置、87...Cレジスタ。

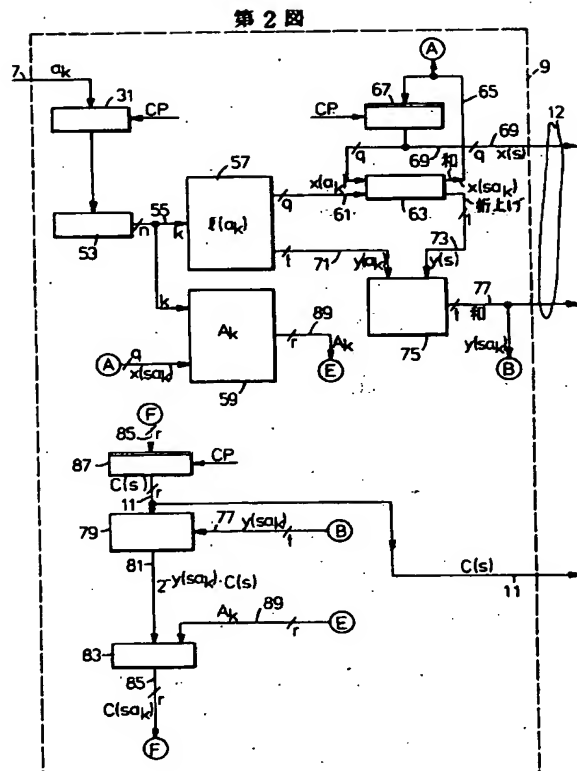


第1A図



第1C図

第1B図



第2図

